4091652

SPERRY RAND

UNIVAC

9400 SYSTEM

Program Description,Drawing

CUSTOMER ENGINEERING

SPERRY RAND UNIVAC

DRAWING NO. 4091652

REVISION A

CUSTOMER ENGINEERING
PRODUCT DIAGNOSTIC SOFTWARE

9400 - Uniscope 300 Test
R3527

	SIGNATURE	TITLE	DATE
APPROVAL		Manager	10/21/70
APPROVAL		Supervisor	10/21/70
APPROVAL		Programming	
APPROVAL		Programming	

REVISION STATISTICS RECORD

DWG REV	MSTR REV	RELEASE DATE	DESIGNER NAME	EIR NUMBER	FCO NUMBER
-	E	8/25/69	O, J. Curl	A89795	A30190024
A	J	11/1/70	P.D.S. Phila. Pa.	C12602	NA

REVISION DESCRIPTION RECORD

REV	DESCRIPTION
-	Release
A	Parameter restrictions for operation with Direct Connection Box Correct Documentation Errors.

TABLE OF CONTENTS

1.0 INTRODUCTION

1.1 Purpose - This document describes the 9400 Maintenance Test Program for the Univac Multi-Station Control Unit (MSCU) Multi-Station Uniscope and the Single Station Uniscope.

1.2 Major Objectives - The test program is designed to perform a Date Line Verification (DLV) test of Remote Uniscopes. The only units affected will be those specifically placed in maintenance program control. Test organization and operator options are intended to satisfy the needs of Design, Quality Control and Field Engineering personnel in check out, acceptance and maintenance activities.

1.3 Equipment Configuration - The Uniscopes are connected to the Date Communication Subsystem (DCS) which in turn is connected to the Multiplexer (MPX) I/O channel of the 9400 Processor.

1.4 Associated Software - This program relies on the Maintenance Control Routine (MCR) to perform program loading, parameterization, execution and I/O handling.

1.5 Reference Documents

Drawing	Rev.	Description
SA 00675		Unimatic Communication System
S 70040		9400 Processor Product Description
4091622		9400 Maintenance Control Routine
4091623		9400 Parameter and Message
4091647		9400 DCS Test

1.6 Restriction - This test is a test of the DATA LINE to either a Mult-Station Control Unit (MSCU) or a Single Station Uniscope.

2.0 FUNCTIONAL DESCRIPTION

This section describes the function and organization of the Uniscope Test Program, which is intended to serve as a Data Line Verification (DLV) Test for Remote Uniscopes.

2.1 General Organization - The test program is designed to output, input and verify data blocks or buffers. The program is devided into the major sections described in the paragraphs which follow.

2.1.1 Parameter - The routines in this section process parameters entered by the operator. Parameter entries may be used to add or delete other parameters or to begin or suspend testing. The View designator allows the user to display the contents of a device table. When parameters are added, the program checks them for correctness; if they are correct, it stores them. If they were not, and error message is printed and the entry is ignored. When parameters are deleted, the affected device is cycled down and its device table is cleared.

Since the Uniscope requires certain control characters, the parameter routine automatically supplies these. This only applies if all of the necessary features are not already present. These features include the following: software sync (F2), EOM (F5), SOM (F3), calculate LRC (F8), and bit length (F1). If the line terminal in question automatically supples any of these, the feature must be deleted by a second or sequential parameter entries. If, however, all of the afore mentioned features are deleted they will automatically be re-entered. This will not happen if even one of the features is present.

2.1.2 Initialize - The programs initialized by biasing addresses within parameter packets and STXIT Packet addresses and then waiting for parameters to be entered.

2.1.3 Uniscope Selection - This routine determines which devices are to be activated by scanning device table until one is found which has been readied by the Parameter routine described in section 2.1.1. After all Uniscopes have been activated and checked when selected, the program starts again at the beginning of the device table. A diagram of device table is shown in Figure 2.

2.1.4 Command - This section is the actual command portion of the program. Commands are issued which enable testing on the selected line terminal, command sequences are issued, status is checked, sense commands are issued, expected sense bytes are calculated, and expected and actual sense bytes are compared. Errors, if detected, are printed. A command table is provided for each command issued in a command test sequence. Figure 3 shows how this table is set up and what it contains.

2.1.5 Data Generation - This routine sets up the proper buffer (1Ø24 Bytes). This routine searches the device table to find the proper RID/DID to be placed in the buffer. This buffer may be either fixed or random data. The routine also calculates output and input byte counts and sets up the proper buffer and status control words.

2.1.6 Interrupt - These routines check the interrupt cue for proper interrupts, for timed-out devices (inactive on erring LT pairs) and updates the cue when a LT pair has been fully tested.

2.17. Data Checking - This routine checks ending status information, process interrupt found in the cue, and verifies the contents of the contents of the input data buffer.

A short explaination of the technique used to check the buffers is in order. The bytes are checked and numbered with respect to the input. This was done in this manner because of the fact that the input is in a reverse order from the output. Figure 5 is a fictitious 15 character uniscope which illustrates the differences in the two buffers.

If one of the special characters which are in the input and not the output buffer is to be checked, it is checked against a program constant. This means that the output buffer is not numbered in a contiguous sequence and is the second reason for number the bytes with respect to the input.

This number also means that the good/bad bytes numbers of the data error message are the number of the bad bytes and its appropriate good byte.

The actual input/output buffers follow this same format but have more characters per line and more lines.

2.2 Subtest Description

2.2.1 Data Subtest -

Objective - The objective of the data subtest is to verify the transfer to and from the remote uniscope.

Method - This is accomplished by transmitting a buffer to the uniscope. The next step is to request a buffer from the same uniscope. Correspondence between the input and output buffers are then checked. The program is subdivided into the major routines shown in Figure 1.

3.0 OPERATING PROCEDURES

3.1 Initialization - Pre-test set up consist of the following:

Load the Maintenance Control Routine (MCR)

3.2 Program Loading - The proceedure for program loading is as follows:

Press the ATTENTION key. The console will respond with an @ sign, the time stamp, a space, and then wait for an input message.

Type in "RU". The console responds by printing "N" adjacent to the "RU" followed by a space.

Type in the program name as follows:

```
R3527
```

Press the EOM key (symbols). The load and run statement appears on the console as:

```
@ 00:00 RUN  R3527 Ⓢ
```

When the program has been loaded, the MCR prints a message which indicates the programs job number and the starting address of the test program.

```
JOB 1, R3527 LOADED AT Ø2B68
```

3.3 Progam Starting - The program is started by the MCR as soon as it is loaded. When the program is started, it types out the message,

```
*A 00:16 1 R3527 ENTER PARAMETERS
```

and waits for a parameter entry. Parameters entered in the format described in paragraph 3.4.1 and using the procedure outlined in paragraph 3.4.

3.4 Program Modifying - The program is adjusted to operate a specific configuration in various operating modes by entering unsoliated parameters.

The proceedure for parameter entry is as follows:

Press the ATTENTION key. The console responds with an @ sign and the time stamp (@ 02:38) and waits for a run identifier entry.

Type in the one-digit job or run number and a comma. A job number entry must be made within two minutes after the ATTENTION key was pressed or an abort typeout will occur. An illustration of an entry to this point is:

```
@ 02:38 1,
```

Enter up to 63 characters of parameter date ending the statement by pressing the EOM key. An operator must enter all data following the comma within two minutes or an abort message will be printed. An illustration of an entry statement is:

@ 02:38 1,A V6/5:#A04 F1/7 Ⓢ

Enter the needed parameters, then begin the program by entering a BEGIN action designator. The begin designator may be entered in two ways: 1) in a seperate statement, or 2) as a seperate sentence in a parameter entry statement.

Three examples are:

@ 00:50 1,B Ⓢ (begin all entered Uniscopes)

@ 00:50 1,B:R20,20/24 Ⓢ (begin selected Uniscope)

@ 00:50 1,A V6/52:#A04,05 F1/7 R20,20/24.B Ⓢ
(begin Uniscope 24 on devices 4 and 5)

3.4.1 Parameter Entries - Parameters are entered in a statement with the following general form:

A Vn/y:#ann Fn/y Rr,r/d,d Ⓢ

Where:

A = an action designator which instructs the program to add the following parameters to the parameter table. Other parameters are described in section.

Vn = number of Variables (program options) to be selected.

Vn/y = an extension which applies to V6 format designates what operator specified data will be stored in the output buffer. If this variable is not specified, random data will be generated.

#ann = arbitrary EBCDIC controller designator, nn indicating the hexidecimal device number or group of numbers (Ø-7F) with an alphanumeric prefix a (A-Z) deniting the controller for the specified devices.

Fn = a feature of features which apply to the device.

Fn/y = an extension which applies to the variables. See section 3.8.2 for detailed description of these extensions.

Rr,r/d,d = (r) indicates the RID characters of the MSCU or SSU (20-7F)$_{16}$.
= (d) indicates the DID characters or Uniscopes (20-37)$_{16}$.

Ⓢ = end of message symbol.

3.4.2 Parameter Notes and Restrictions

1. Entering the action designator Dⓢ or D:#A40ⓢ will delete the device pair and all RID (s) and DID (s) associated with the device pair. Examples are as follows:

```
@ 02:35 1,Dⓢ

@ 02:35 1,D:#A40,41ⓢ

@ 02:35 1,D:#A40ⓢ
```

2. To Delete specifiec RID(s) or DID(s), only the RID(s) or DID(s) need be entered. If a RID is entered without a DID entry, all DID(s) associated with that RID will also be deleted. Examples are as follows:

```
@ 02:36 1,D:R20,20/21ⓢ (deletes a DID 21)

@ 02:36 1,D:R20,20/21-27ⓢ (deletes DIDs 21-27)

@ 02:36 1,D:R20,20ⓢ (deletes RID 20,20 and all associated DID(s) )
```

3. Entering the action designator Eⓢ, E:#A40,41 or E:#A40 will suspend the entire test program. Examples are as follows:

```
@ 02:37 1,Eⓢ

@ 02:37 1,E:#A40ⓢ

@ 02:37 1,E:#A40,41ⓢ
```

4. To END specific RID(s) or DID(s), only the RID(s) or DID(s) need be entered. If a RID is entered without a DID entry, all DID(s) associated with that RID will also be suspended. Examples are as follows:

```
@ 02:38  1,E:R20,20/21 ⓢ (suspends DID 21)

@ 02:38  1,E:R20,20/20-27ⓢ (suspends DID(s) 21-27)

@ 02:38  1,ER20,20ⓢ (suspends RID 20,20 and all associated DID(s) )
```

5. A Delete causes a device to be cancelled and its device table to be cleand. The device can only be reactivated by adding it again. An End causes the device to suspend but its device table is not cleared. The device may be reactivated by using a BEGIN designator.

6. A colon (:) is used to seperate the controller and device fields in a parameter statement.

7. A semicolon (;) is used to seperate different kinds and associated features having the same controller field within one parameter sentence.

@ 00:50 1,A V6/52:D40 F1/2 R20,20;R21,22/28 Ⓢ

8. A period is used to seperate parameter sentences.

@ 00:50 1,A V6/52:D40 F1.D:D30 Ⓢ

9. Entering the action designator 1,B Ⓢ will activate all devices previously selected but not activated.

3.5 Program Stopping - The test program can be stopped by one of the following:

Via a parameter entry having the following format:

E Ⓢ

This entry will cause the test program to suspend without a console indicator.

Via a parameter entry having the following format:

D Ⓢ

This entry will cause the test program to suspend and clear all device tables.

3.6 Program Restarting - The test program can be restarted by the following parameter entry:

B Ⓢ

This entry will restart the test program from its initial starting point.

3.7 Program Termination - The program may be removed from storage by submitting a cancel directive to the Maintenance Control Routine as follows:

1. Press the ATTENTION key. The MCR will type out the time and wait for another entry.

2. Type the letters "CA", the MCR will respond with the letters "NCEL" and a space. A typeout to this point appears as:

00:50 CANCEL

3. Type the job indentifier and stop code.

00:50 CANCEL 1 Ⓢ

3.8 Program Designators - The parameter designators recognized by this test program are grouped into the following three categories:

Action Designators

Equipment Designators

Program Designators

3.8.1 Action Designators - The Action Designator specify how the Parameter analysis Routine of the test program process the desired parameters. The following Action Designators are recognized by the test program:

A = add D = delete V =View
B = begin E = suspend

3.8.2 Equipment Designator - Equipment Designators define the particular device being tested.

Ann = A three-character device address consisting of, in order, the alpha character (arbitrarily assigned) which designates the subsystem's control unit and a two-digit hexadecimal number (ØØ-7F) which specifies a selected device (unit) in the subsystem.

Fn = A decimal number (1-14) which specifies a hardware feature, as follows:

F1/n = Bit Length Feature where n defines the bit length.

F2/n = Synchronization Character Feature for synchronous line terminals where n defines the character. Software Sync. character.

F3/n = Software Output Start-of-Message (SOM) Character Feature where n defines the SOM character.

F4/n = Software Output End-of-Transmission (EOT) Character Feature for low-speed line terminals where n defines the EOT character.

F5/n = Software Output End-of-Message (EOM) Character Feature where m defines the EOM character.

F7 = Pause Feature where a pause character is recognized by the line terminal.

F8/n = Longitudinal Redundancy Check (LRC) Feature where LRC is generat by the DCS according to n, as follows:

n = 0, LRC present but not to be verified
n = 1, SOM through EOM generation
n = 2, SOM - last data character and F1005-04/05
n = 3, First data character - EOM

F9 = Automatic Look-for-Sync (LFS) Feature where a synchronous line terminal automatically looks for synchronization after successful completion of the previous transfer or issues a command to permit the input to look for synchronization. EOM recognition must occur in order to maintain automatic LFS.

F11 = Software Input Start-of-Message (SOM) Feature if SOM differs from that specified in F3.

F12 = Software Input EOM if different from the output or the hardware.

F13 = Dialing Adapter Toggle Inhibit Feature. When it is set, no togglin occurs between the Dialing Adapter Test and the Data test or its associated LT.

F14 = Dialing Adapter Present Feature.

3.8.3 Program Designators - Program designators are parameters which modify the test program testing procedures. The following designators are allowed for this test program.

VN = A number indicating what variables are to apply to the subsystem.

V1 = Allow Error Printout (available on load)

V4 = Command and Interrupt Trace (Not for use with Direction Connection Box)

V6 = Operator specified data

V7 = Direct Error message to High Speed printer

V20 = Enable Buffer Printout on Data Error

3.9 Message Description -

3.9.1 General - When the uniscope test program detects an abnormal situation, it informs the operator via a console message. If vaiable seven (V7) was entered and a printer is available the messages will be displayed on the printer.

3.9.2 General Format Information - The mnemonics and symbols used in the message described in 3.9.3 are defined in the following:

3.9.2.1 Mnemonics -

CC	=	Current Command
DPC	=	Device Previous Command
CPC	=	Controller Previous Command
RCVCC	=	Received Condition Code
CS	=	Current Status
PS	=	Previous Status

ES	=	Expected Status
AE	=	Address Expected
AR	=	Address Received
D	=	Declarative Message
I	=	Imperative Message
Q	=	Question
SB	=	Sense Bytes
MB	=	Monitor Sense Bytes
GB	=	Good Bytes
BB	=	Bad Bytes
TBR	=	Total Bytes Received
TBB	=	Total Bad Bytes
TBE	=	Total Bytes Expected
Pn	=	Pattern Number
T	=	Subtest
D	=	Device
R3527	=	Program Name (Uniscope 3527)

3.9.2.2 Symbols -

hh	=	Hour (00:23)
MM	=	Minute (00:59)
#	=	Buffer position of data word or byte
r	=	Run number
n or nn	=	Numerical Value of Mnemonic prefix

3.9.3 Messages - All messages originating from the test program fall into three groups: Parameter Error, Subsystem Error, and Information Messages.

3.9.3.1 Parameter Error Messages - Parameter error messages are printed on the console insufficient or incorrect parameters are detected by the test program. The following is a list of all Parameter Error Messages:

1. Lacks Device

   Cause: The program has no device to test or all devices have been deleted or suspended.

   Program Action: The program prints the following message and waits for the operator to enter parameters.

   Example: D hh:mm 1 R3527 LACKS DEVICE

   Operator Action: Enter parameters making certain to specify a device.

2. Parameter Format Error

   Cause: Parameters were not entered in the correct format.

   Program Action: The program prints the following message and waits for a correct parameter entry.

   Example: D hh:mm 1 R3527 PARAM FORMAT ERR

   Operator Action: Re-enter parameters correctly.

3.9.3.2 Subsystem Error Messages - Subsystem error messages are printed when incorrect status or sense bytes are detected, when unexpected timeout errors occur, or when incorrect data is received. Subsystem error messages are also printed when the number of bytes transferred was incorrect. Subsystem error messages are printed in one of the following formats:

1. Status Error

Cause: A status error has occurred.

Program Action: Program sends a Sense Command to the affected LT only if a Unit Check status indication was returned from the line terminal.

Example:

```
D hh:mm 1 R3527 STATUS ERROR-DEVICE #ann T1/n
CC = nn  CS = nn  ES = nn
```

2. Sense Error

Cause: The program did not get correct immediate status.

Program Action: The program readies the device for restarting.

Example:

```
D hh:mm 1 R3527 SENSE ERROR-DEVICE #ann T1/n
CC = nn  CS = nn  ES = nn
```

3. Unexpected Interrupt

Cause: An interruption has occurred from an inactive or non-existent device.

Program Action: The program processes the next interruption in the cue

Example:

```
D hh:mm 1 R3527 UNEX INTERRUPT DEVICE #ann
CS = nn T1/n
```

4. Time-out Error

Cause: A device has timed out (i.e. = the time-out word in the device table indicates that operating time for the device has exceeded 15 seconds.)

Program Action: The program clears the Active and I/O Device Flags and checks if the next active device has timed out. The device is also set for re-activation.

Example:

```
D hh:mm 1 R3527 TIMEOUT ERROR DEVICE #ann T1/n.
```

5. Data Error

Cause: Verification of the output and input data buffers indicated a data error.

Program Action: The program reactivates the device.

Example:

```
D hh:mm 1 R3527 #ann RID=n,m DID=n DATA ERROR
GB Ø = nn 1 = nn 2 = nn ------n = nn
BB Ø = nn 1= nn 2 = nn ......n = nn

TB = nn
```

6. Bad Sense Status in Cue

Cause: An interruption other than Device End/Channel End.

Program Action: The cue is updated and the device is re-activated.

Example:

```
D hh:mm 1 R3527 BAD SENSE STATUS IN CUE DEVICE #ann
ES = ØC  CS = nn                                       T1/n
```

7. Bad Status in Cue

Cause: A Device End/Channel End status indication was not received as expected.

Program Action: The cue is updated and the device is re-activated.

Example:

```
D hh:mm 1 R3527 BAD STATUS IN CUE-DEVICE #ann T1/n
ES = ØC  CS = nn
```

8. Bad Sense Bytes

Cause: Bad Sense Bytes have been received in responce to to command.

Program Action: The cue is updated and the next device is activated.

Example:

```
Dhh:mm 1 R3527 BAD SENSE BYTES DEVICE #ann T1/n
```

9. Condition Code Error on SIO

Cause: A condition code error has occured during ans SIO (Start I/O) operation.

Program Action: The cue is updated and the next device is activated.

Example:

```
D hh:mm 1 R3527 COND CODE ERROR ON SIO
COND CODE = nn  DEVICE #ann
CC = nn CS = nn  ES = nn
```

10. Remote Error

Cause: Improper response from Uniscope has been encountered.

Program Action: The uniscope is polled again

Example:

```
D hh:mm 1 R3527 Device #ann RID=m,n DID=n
Status Error From Remote
```

3.9.3.3 Information Messages - Information messages are printed whenever it becomes necessary for the test program to inform the operator that a condition exists. The following is a list of all information messages:

1. Enter Parameters

   Cause: The test program was just loaded by the MCR.

   Program Action: The program prints the following message, waits for parameters, and proceeds when it encounters B s..

   Example: D hh:mm 1 R3527 ENTER PARAMETERS

   Operator Action: Enter parameters for the test program.

2. Sense Bytes

   Cause: When a status error occurs, the program sends a Sense Command and prints the sense bytes received.

   Program Action: The cue is updated and the device is re-activated.

   Example: D hh:mm 1 R3527 SENSE BYTES-DEVICE #ann T1/n
   SB = nnnn

3. Aborted - One at a time, Guys

   Cause: Operator has tried to load the program twice without cancelling the first load or has attempted to load another real-time program.

   Program Action: The second load attempt is ignored.

   Example: ABORTED-ONE AT A TIME, GUYS

4. Device Not Available

   Cause: A device which was addressed by the program is off-line.

   Program Action: The program re-activates the device.

   Example: D R3527 DEVICE #ann NOT AVAILABLE

5. Command Trace

   Cause: Variable 4 has been enabled.

   Program Action: The program prints the following message every time a command is issued to the DCS.

   Example: D hh:mm 1 R3527 CMD TRACE #ann
   CC = nn CS = nn ES = nn T1/n

6. Interrupt Trace

Cause: Vaiable 4 has been enabled.

Program Action: The program prints the following message whenever an interrupt is received from the DCS.

Example:

```
INT TRACE INT = nnnnnnnn  PS = nn  BCWØ = nnnnnnnn
BCWI = nnnnnnnn T1/n
```

7. View

Cause: The operator has entered a View Parameter.

Program Action: The program prints out the contents of the device table for the selected device.

Example:

```
D hh:mm 1 R3527 # ann nnnnnnnn  nnnnnnnn
nnnnnnnn  nnnnnnnn  nnnnnnnn  nnnnnnnn
nnnnnnnn  nnnnnnnn  nnnnnnnn  nnnnnnnn
```

8. Buffer Dump

Cause: Variable 20 has been enabled

Program Action: The program prints the aollowing message whenever a data error is encountered

Example:

```
D hh:mm 1 R3527 DEVICE #ahn RID=m,n DID=n Buffers
R3527 OUTPUT Buffer
nnnnnnnn  nnnnnnnn  nnnnnnnn  nnnnnnnn
nnnnnnnn  nnnnnnnn  nnnnnnnn  nnnnnnnn
R3527 INPUT Buffer
nnnnnnnn  nnnnnnnn  nnnnnnnn  nnnnnnnn
nnnnnnnn  nnnnnnnn  nnnnnnnn  nnnnnnnn
```

9. Table Exceeded

Cause: The operator has entered a RID/DID and the maximum number has already been entered.

Program Action: The entry is rejected and the program continues running the entered Uniscope. Uniscope (RID/DID) must be deleted to enable the entry.

Example:

```
D hh:mm 1 R3527 DEVICE RID/DID
Table full
```

4.0 SUPPLEMENTARY INFORMATION

4.1 Supplementary Software Information - This paragraph contains information supplementary to the information in Section 1 through 3. Figure 1 is a block diagram of the test program. Figure 2 (and Tables 1 ans 2 defines the content and format of the device tables. Figure 4 shows the data buffer used by the Data Test.

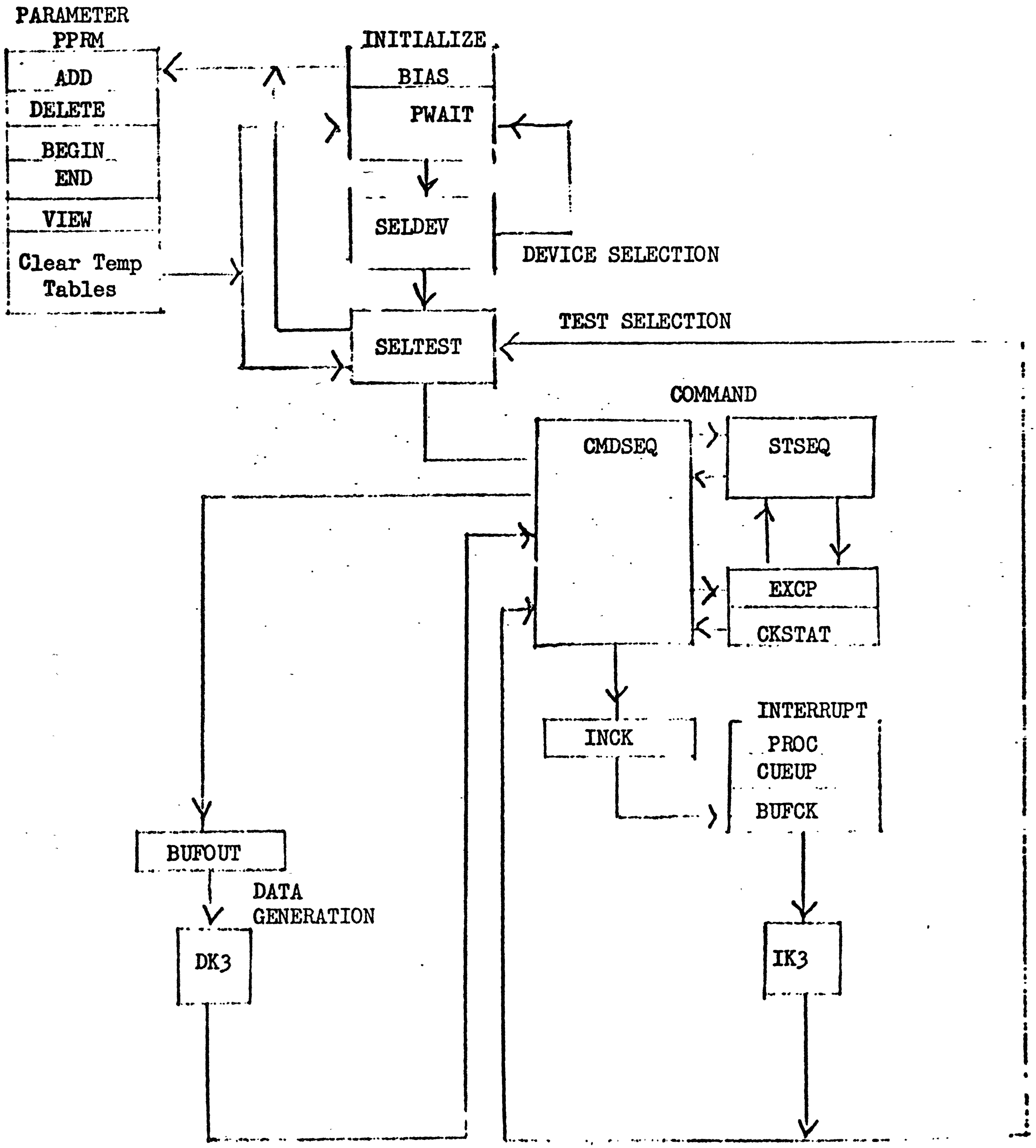


FIGURE 1

4.1. Supplementary Software Information (Continued) -

Byte				
Ø	See Table 1	Output Device no.	F1 Bit Length	F2 Sunc
4	F3 Output SOM	F4 EOT	F5 Output EOM	F7 Pause
8	F11 Input SOM	F8 LRC Type	F12 Input EOM	Subtest
12	Buffer Pointer	Kind	See Table 3	
16	Not Used	Controller Alpha	Not Used	Time-out Counter
2Ø	RID Character 1	RID Character 2	See Table 2	DID
24	See Table 2	DID	RID Character 1	RID Character 2
256	RID Character 1	RID Character 2	SEE Table 2	DID

Table 1 - (Byte Ø of Device Table)

BIT	FUNCTION
Ø	Device Active
1	Device Selected
2	Device Suspended
3	Output Interrupt Received
4	Output Sense
5	Output Sent
6	Input Interrupt Received
7	Input Sense Command Sent

Table 2 - (Byte Preceeding Each DID)

BIT	FUNCTION
Ø	DID Flag
1	DID Select
2	DID Suspended
5	DID Tested

FIGURE 2

4.1 Supplementary Software Information (Continued) -

Table 2 - (Bytes 14 and 15)

BYTE 14

BIT	Fn	FUNCTION
Ø		Used as DA flag for Dialing Adapter
1	F1	F1 Bit Length Enable
2	F2	F2 Software Sync Enable
3	F3	F3 Software Output SOM Enable
4	F4	F4 Software EOT Enable
5	F5	F5 Software Output EOM Enable
6	—	Not Used
7	F7	F7 Pause Generation Inhibit Enable (Random Data)

BYTE 15

BIT	Fn	FUNCTION
Ø	F8	F8 LRC Type Enable
1	F9	F9 Automatic Look-for-Sync Enable
2	F10	F10 Local Test Enable
3	F11	F11 Software Input SOM Enable
4	F12	F12 Hardware EOM or Software Input EOM
5	F13	F13 Dialing Adapter Toggle Inhibit
6	F14	F14 Dial Adapter Present
7	—	Dialing Adapter Toggle Flag

The Longitudinal Redundancy Check (LRC) Type Enable (Bit Ø of Byte 15) specifies:

Ø = LRC is present but not to be verified
1 = SOM EOM
2 = SOM DATA
3 = DATA EOM

Note that F9 and F6 are not used at this time. In addition, it is necessary to use F10 and F12 only when the input differs from the software generated output.

Command	Immediate Status	Sense Bytes								Sense Bytes							
Expected Status	Not Used	Ø	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Ø - not used

1 - General poll enable

2 - data compare enable

3 - generate data and data BCW's and SCW's

4 - sense byte output compare enable

5 - sense byte input compare enable

6 - ship the command if previous command was in error

7 - exit test on error

8 - interrupt is experted from this command

9 - input Device Command

10 - repeat command twice before looking at status

11 - exit after command to wait for interrupt

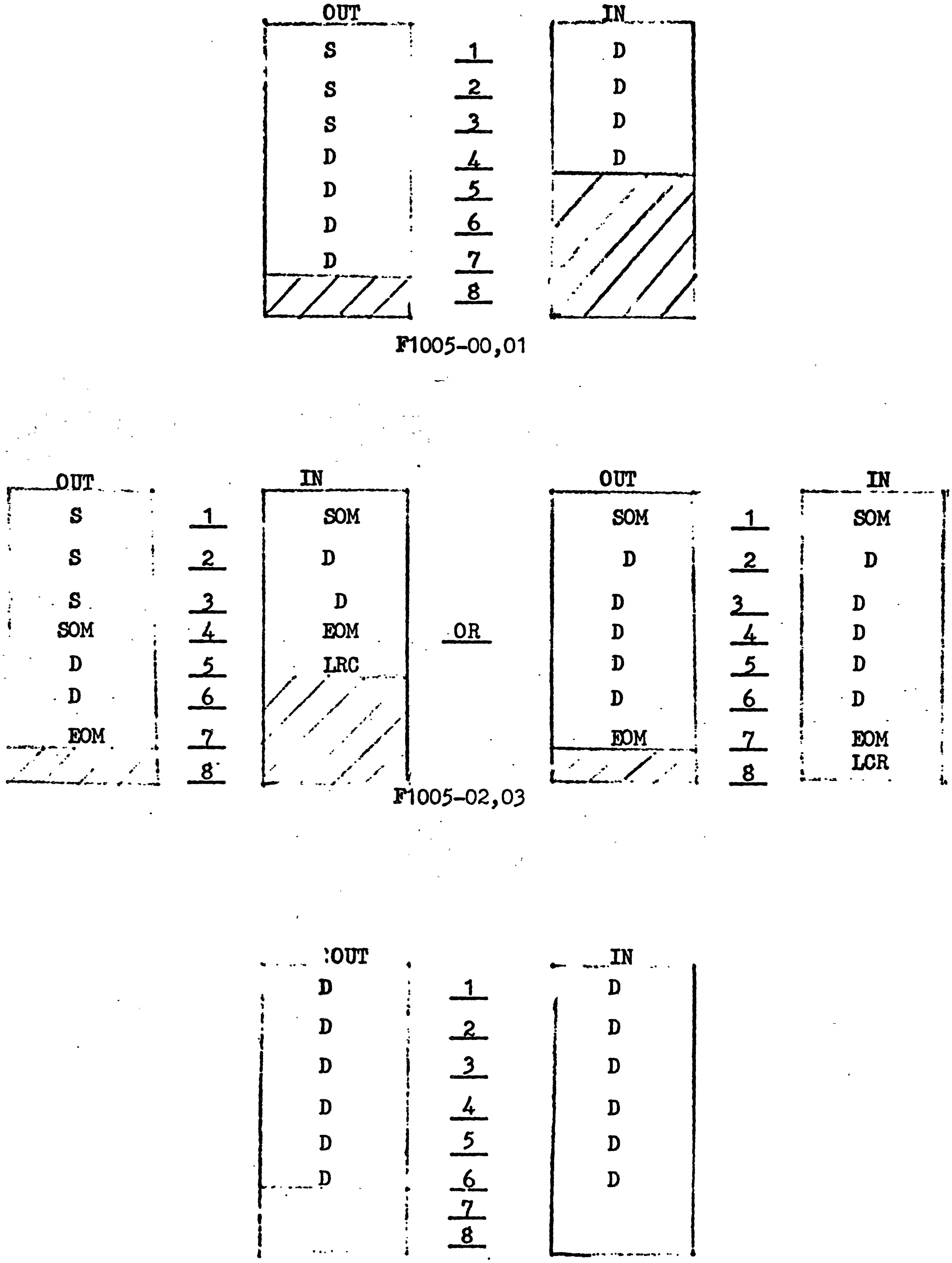
12 - store sense byte for compare

13 - poll command

14 - initialize Before Command

15 - last command

FIGURE 3



F1005-00,01

F1005-02,03

F1005,04,05

FIGURE 4. Synchronous LT Data Buffers

OUTPUT BUFFER

	Sync (1)
	Sync (2)
	Sync (3)
	SOM
	RID 1
	RID 2
	DID
	Output Code Ø7
	Cursor Command
	Cursor Character
	Cursor Line
line 1 character 1	Text (1)
line 1 character 2	Text (2)
line 1 character 3	Text (3)
line 1 character 4	Text (4)
line 1 character 5	Text (5)
line 2 character 1	Text (6)
line 2 character 2	Text (7)
line 2 character 3	Text (8)
line 2 character 4	Text (9)
line 2 character 5	Text (10)
line 3 character 1	Text (11)
line 3 character 2	Text (12)
line 3 character 3	Text (13)
line 3 character 4	Text (14)
line 3 character 5	Text (15)
	EOM
	LRC
	EOT

INPUT BUFFER

	SOM
	RID
	RID
	DID
	Poll Acknowledge (Ø6)
	Uniscope SOM (Ø4)
line 3 character 5	Text (15)
line 3 character 4	Text (14)
line 3 character 3	Text (13)
line 3 character 2	Text (12)
line 3 character 1	Text (11)
	Cursor Return (ØB)
line 2 character 5	Text (10)
line 2 character 4	Text (9)
line 2 character 3	Text (8)
line 2 character 2	Text (7)
line 2 character 1	Text (6)
	Cursor (ØB)
line 1 character 5	Text (5)
line 1 character 4	Text (4)
line 1 character 3	Text (3)
line 1 character 2	Text (2)
line 1 character 1	Text (1)
	Cursor Position
	Cursor Character
	Cursor Line
	EOM
	LRC
	EOT

Text (1) Line (1) Character (1)	Text (2) Line (1) Character (2)	Text (3) Line (1) Character (3)	Text (4) Line (1) Character (4)	Text (5) Line (1) Character (5)
Text (6) Line (2) Character (1)	Text (7) Line (2) Character (2)	Text (8) Line (2) Character (3)	Text (9) Line (2) Character (4)	Text (10) Line (2) Character (5)
Text (11) Line (3) Character (1)	Text (12) Line (3) Character (2)	Text (13) Line (3) Character (3)	Text (14) Line (3) Character (4)	Text (15) Line (3) Character (5)

UNISCOPE SCREEN

Figure 5